

Treball de Fi de Grau

## **Grau en Enginyeria en Tecnologies Industrials**

# **Creació d'una GUI per a l'anàlisi de les notes de la fase inicial del GETI**

**MEMÒRIA**

**Autor:** Marc Vila Ribera

**Director/s:** Lluís Solano

**Convocatòria:** Juliol 2016



Escola Tècnica Superior  
d'Enginyeria Industrial de Barcelona



## Resum

En aquest projecte s'ha creat una Interfície Gràfica d'Usuari (GUI) per visualitzar l'anàlisi estadístic d'una base de dades. L'anàlisi s'ha fet sobre el conjunt de notes dels estudiants del Grau en Enginyeria en Tecnologies Industrials durant la seva fase inicial. En aquest treball s'ha comparat aquestes notes amb la nota d'accés de cada un dels alumnes.

Per fer aquesta anàlisi s'han fet 4 estudis diferents:

- Estadístiques de cada assignatura
- Gràfica de les notes de l'assignatura en funció de la nota d'accés
- Assignatures aprovades durant un any acadèmic
- Estimació de notes en funció de la nota d'accés

Aquests estudis s'han volgut visualitzar intentant oferir una fàcil interpretació dels resultats i una flexibilitat en possibles estudis posteriors. És per això que s'ha creat una interfície gràfica d'usuari mitjançant el model vista controlador; un mètode que permet crear i governar una interfície d'usuari de manera molt clara.

Per crear aquesta GUI s'ha gaudit de la facilitat de la multiplataforma Qt, que permet de programar-la amb el llenguatge informàtic Python.

Pel que fa a l'anàlisi de dades, s'han usat els mòduls Pandas i Matplotlib per ajudar al tractament de les dades, l'obtenció de dades estadístiques i la creació de gràfiques.

# Sumari

<b>RESUM</b>	<b>2</b>
<b>SUMARI</b>	<b>3</b>
<b>1. INTRODUCCIÓ</b>	<b>5</b>
1.1. Motivació .....	5
1.2. Requeriments previs .....	5
<b>2. OBJECTIUS I ABAST DEL PROJECTE</b>	<b>7</b>
2.1. Objectius del projecte.....	7
2.2. Abast del projecte .....	7
<b>3. ESTRUCTURA GLOBAL DEL PROJECTE</b>	<b>8</b>
3.1. Metodologia àgil .....	10
3.2. Eines .....	10
3.2.1. Pandas.....	10
3.2.2. Matplotlib.....	11
<b>4. ANÀLISI DE DADES</b>	<b>12</b>
4.1. Obtenció de dades .....	12
4.2. Filtratge.....	12
4.3. Presentació de les dades.....	14
<b>5. INTERFÍCIE GRÀFICA</b>	<b>15</b>
5.1. PyQt .....	15
5.2. Qt Designer .....	16
5.3. Model vista controlador .....	17
<b>6. ESTUDIS</b>	<b>21</b>
6.1. Estadístiques de les assignatures .....	21
6.2. Gràfica notes vs selectivitat .....	24
6.3. Assignatures aprovades per any acadèmic.....	26
6.4. Predicció.....	29
<b>7. IMPACTE AMBIENTAL</b>	<b>33</b>
<b>8. PLANIFICACIÓ I VALORACIÓ ECONÒMICA</b>	<b>34</b>
8.1. Planificació .....	34
8.2. Valoració econòmica.....	35

<b>CONCLUSIONS</b>	<b>36</b>
<b>AGRAÏMENTS</b>	<b>38</b>
<b>BIBLIOGRAFIA</b>	<b>39</b>

# 1. Introducció

## 1.1. Motivació

La tria d'aquest projecte ha estat a causa de diversos motius. Per una banda, la poca informàtica que es tracta durant el Grau en Enginyeria en Tecnologies Industrials ha estat suficient per despertar-me una curiositat i una motivació per la informàtica. I precisament perquè hem tractat poc amb la informàtica tinc aquest desig d'aprendre'n més i adquirir nous coneixements.

Per altra banda, quan entro a l'Escola el 2012 em trobo una sèrie de gent que, tot i tenir una nota altíssima a la selectivitat, li costa molt aprovar algunes assignatures del Grau. A més, n'hi havia que no semblaven tan brillants i, en canvi, tiraven endavant sense grans esforços. Aquesta curiositat em va fer pensar si realment la selectivitat és una bona prova d'accés i aquest projecte m'ha permès de comprovar aquest dubte.

## 1.2. Requeriments previs

Per tal de realitzar aquest projecte han estat necessaris uns coneixements informàtics previs.

Per la banda de l'anàlisi de dades ha estat essencial informar-se i practicar amb les llibreries usades, sobretot les de Pandas i Matplotlib, per poder saber les opcions que ofereixen aquestes llibreries tan àmplies. Tot i que només ha estat necessària una part de les llibreries, s'ha hagut de profunditzar per poder entendre quina era la millor opció pel projecte que s'ha dut a terme.

Dins del que és la part de visualització de resultat s'ha usat la interfície gràfica PyQt i, com és evident, es requereix d'uns coneixements per poder utilitzar-lo. Per tal de fer-ho més entenedor, el paquet Qt ofereix una opció per al PyQt que s'anomena QtDesigner, del qual també ha estat fonamental documentar-se per poder començar a usar aquesta eina.



## **2. Objectius i abast del projecte**

### **2.1. Objectius del projecte**

Els objectius generals del projecte són, per una banda, fer un estudi de les notes de l'alumnat de l'escola durant la seva fase inicial i, per l'altra, crear una interfície gràfica per poder visualitzar els resultats d'aquesta anàlisi realitzada prèviament.

Profunditzant més en el treball, els objectius més específics són els següents:

- Fer càlculs estadístics de les notes dels alumnes per a cada assignatura
- Descobrir quants són els estudiants que ho aproven tot
- Predir resultats de la fase inicial en funció de la nota d'accés
- Visualitzar la nota de cada assignatura en funció de la nota de selectivitat de cada alumne
- Crear una interfície gràfica còmode i entenedora per visualitzar els resultats

### **2.2. Abast del projecte**

Aquest projecte s'ha centrat en els estudiants del Grau en Enginyeria en Tecnologies Industrials (GETI) de l'Escola Tècnica Superior Enginyeria Industrial de Barcelona. A més, s'ha focalitzat l'estudi en la fase inicial del Grau, és a dir, tan sols el primer any. També s'ha considerat, a l'hora de seleccionar les notes dels estudiants, només la primera convocatòria a la que els alumnes s'han presentat. Per tant, si s'han tornat a presentar més tard perquè no han aprovat l'assignatura no s'ha comptabilitzat per l'estudi del projecte.

### 3. Estructura global del projecte

Per fer un projecte senzill d'entendre i còmode de treballar-hi és fonamental l'estructura que en què es basa el treball. Primerament es crearà un directori de treball, des d'on es realitzarà tot el projecte, per tal de minimitzar la complicació innecessària de tenir diversos espais de treball i haver de canviar d'un a l'altre. Dins d'aquest directori, es treballarà en els dos camps que cobreix el projecte: l'anàlisi de dades i la creació de la interfície gràfica. Es mantindran aquests dos projectes al mateix espai de treball ja que la interfície gràfica rebrà les dades dels fitxers d'anàlisi i així es simplifica la feina.

Un cop establert l'espai de treball, l'estructura del projecte es dividirà en aquest dos camps que es comentava. Per una banda, per l'anàlisi de dades es disposarà del document que conté les dades des del principi i, a mesura que avanci el treball s'anirà modelitzant el fitxer usat per tractar-les, en funció dels estudis que es vagin realitzant.

Si ens fixem en aquesta part del treball, es tindran dos fitxer: un de dades i un que, usant les dades del fitxer anterior, fa les anàlisis requerides pel projecte. Com que el segon fitxer necessita del primer les dades que conté podem dir que estan directament relacionats.

El fitxer de dades és un document que estarà prèviament filtrat segons l'abast del nostre estudi, és a dir, notes d'alumnes del GETI durant la fase inicial, a més d'altres filtratges més exhaustius que s'explicaran més endavant. Aquestes dades, un cop filtrades es prepararan per ser analitzades amb eines com el Pandas o el Matplotlib que també s'expliquen seguidament.

Un cop es tinguin les dades preparades, es procedirà a fer l'anàlisi. Responent als objectius del projecte, es plantejaran quatre estudis:

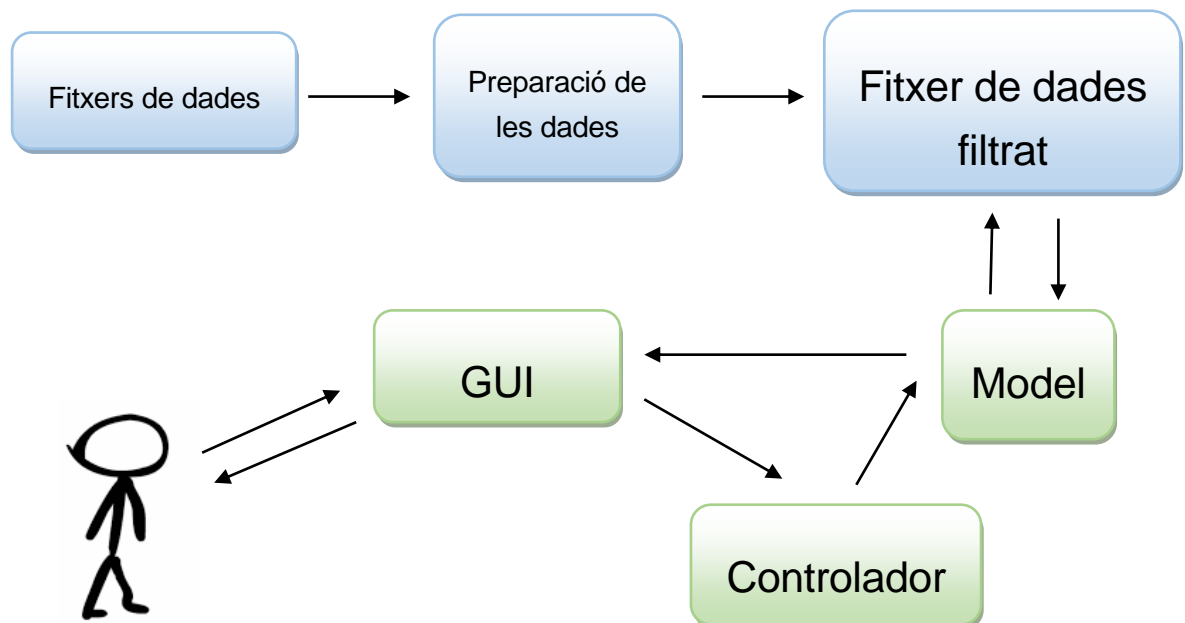
- **Estadístiques de cada assignatura.** Per aquest estudi es determinarà, per a cada assignatura: quina és la seva mitjana, com és la distribució de notes (mitjançant els quartils) i altres mesures estadístiques com el màxim i el mínim. A més de fer-ho per a cada assignatura també es tindrà en compte l'any que es selecciona.
- **Notes de l'assignatura en funció de la nota d'accés.** Per determinar la nota de cada assignatura en funció de la nota d'accés de cada alumne es farà, igual que en l'estudi anterior, separant per curs acadèmic. A partir del fitxer de dades filtrades s'agafaran els alumnes que hagin cursat l'assignatura aquell any i se'n farà una gràfica on les notes de l'assignatura es mostraran a l'eix vertical i les notes d'accés a l'eix horitzontal.



- **Quantes assignatures aprova cada alumne?** Quant al tercer estudi, es vol determinar les assignatures que aproven els alumnes durant un any acadèmic. Per fer-ho, només es tindrà en compte el primer any acadèmic de cada alumne a l'escola i es determinarà quantes assignatures té aprovades durant aquell termini.
- **Estimació de notes en funció de la nota d'accés.** Com que hi ha moltes maneres de fer una estimació o predicció futura, s'ha hagut de decidir un criteri des d'un bon principi. Per fer aquesta estimació de notes futures en funció de la nota d'accés es classificaran els alumnes en rangs de notes d'accés de mig punt. Per a cada un dels alumnes que sigui dins de l'interval s'agafaran les notes de cada assignatura i es farà una nota mitjana amb tots els altres alumnes del mateix rang d'estudi. Aquestes notes mitjanes seran les que es donaran com a predicció per a altres alumnes.

Per altra banda, paral·lelament es treballarà amb la interfície gràfica d'usuari (GUI). Aquesta part està dedicada a visualitzar les dades que es vol de manera clara i fàcil d'interpretar. Dins d'aquest bloc del treball també se'n poden diferenciar dues parts: la pròpia interfície, que és el que nosaltres podem veure i amb la qual podem interactuar; i el fitxer que fa d'enllaç entre la interfície i l'anàlisi de dades, el controlador.

Per implementar la GUI es farà servir el model vista controlador. Aquest mètode és molt eficaç i proporciona una organització del treball que redueix molt la complexitat. El model vista controlador permetrà de separar molt bé els blocs de treball anomenats prèviament: la vista, el controlador i el model, que és l'encarregat de fer l'anàlisi de dades.



### 3.1. Metodologia àgil

Per tirar endavant el projecte, el dia a dia s'ha basat en el que s'anomena una metodologia àgil. Aquesta metodologia basa el seu funcionament en l'adaptabilitat als canvis. I és que en la informàtica això és una constant ja que si no es fa correctament una cosa, la resta ve a sobre i et pot comportar problemes posteriors, per tant, s'ha de fer un procés iteratiu en tasques petites, fer-les bé cada una i assegurar-se'n abans de passar al següent pas.

Això és el que s'ha fet en aquest projecte. Es podria haver començat a fer tota la UI, tirar endavant i acabar-la abans de fer l'anàlisi de dades i, un cop fos acabada, començar amb l'anàlisi i visualitzar-lo a la UI. O bé al revés, fer tot l'anàlisi de dades i, quan s'hagués finalitzat, començar a crear la UI per visualitzar l'estudi. En lloc de tot això, s'ha anat fent pas a pas les dues coses paral·lelament. Per clarificar el seu funcionament, cada estudi s'ha fet separatament, de manera que un cop s'ha fet l'anàlisi de dades per l'estudi, seguidament s'ha procedit a combinar l'estudi amb la UI per tal de visualitzar-lo i poder detectar els possibles errors a mesura que es va avançant en el projecte.

Aquesta metodologia ha permès de, fent canvis petits quan toca, poder fer un mètode de prova i error. Per contra, si no s'hagués adoptat aquesta manera de fer el projecte es podria haver trobat un error al final del treball i haver de canviar-ho tot.

A més, la metodologia àgil defensa el canvi, per sobre de traçar un pla tancat des del principi, depenent de les necessitats del client, de les interaccions del programari amb l'usuari i posa per davant la funcionalitat per sobre de la documentació exhaustiva. Aquesta també és la idea que s'ha tingut durant tot el projecte: posar l'usuari i la funcionalitat per davant d'un pla tancat i adaptar el treball a les condicions i problemes que puguin sorgir.

Aquesta metodologia, per raons pràctiques, s'ha estès molt sobretot en projectes de software i actualment és utilitzada arreu.

### 3.2. Eines

#### 3.2.1. Pandas

Aquest és un mòdul de Python destinat principalment a l'anàlisi estadística de dades. Per tant, no hi ha millor cabuda per aquest mòdul que aquest treball. El format més comú de presentació de dades per aquest mòdul és el DataFrame i és fàcilment assolible a partir d'un fitxer csv, d'aquí la importància de l'apartat 4.3.

### 3.2.2. Matplotlib

Matplotlib és una llibreria de Python que permet generar gràfiques. A més, és una llibreria molt fàcil d'utilitzar i molt documentada que permet l'ús tant per a principiants com per estudis de primer nivell. Les possibilitats són molt àmplies i van des de gràfiques *scatterplots* com els que s'usaran aquí fins a gràfiques 3D passant per espirals logarítmics i tot tipus de gràfiques.

## 4. Anàlisi de dades

### 4.1. Obtenció de dades

Per l'obtenció de dades s'ha recorregut als fitxers de dades de la plataforma Prisma de l'escola, que inclouen les qualificacions i algunes dades personals de tots els estudiants que cursen tant el Grau en Enginyeria en Tecnologies Industrials (GETI), com el Grau en Enginyeria Química (GEQ) i el Grau en Enginyeria de Materials (GEM).

Gràcies a aquesta plataforma es van aconseguir 3 fitxers: el fitxer de dades personals de cada estudiant, el fitxer de notes de la fase inicial i el de notes de cursos posteriors.

- Fitxer de dades personals de l'estudiant: Conté el codi del programa al que estan matriculats (GETI, GEQ o GEM), el codi de l'estudiant, el sexe, el codi postal, l'any d'accés, el tipus d'accés, la nota d'accés, i el centre del que provenen.
- Fitxer de notes de la fase inicial: Essent cada fila del fitxer una assignatura matriculada de 1r curs, el fitxer conté el codi de l'estudiant, el programa que cursa, el codi de l'assignatura, el curs i el quadrimestre en què la cursa, la nota que té i a quin grup està matriculat.
- Fitxer de notes de cursos posteriors: Aquest fitxer té el mateix format que el de notes de la fase inicial i les dades que conté són les mateixes, però amb les assignatures corresponents al 2n, 3r i 4t curs.

### 4.2. Filtratge

Un cop en possessió d'aquestes dades, una part molt important ha estat la selecció de les dades rellevants per a l'estudi per tal de poder filtrar-les i no haver de treballar sempre amb tot el fitxer.

A més, cal destacar que els fitxers extrets de la plataforma Prisma no estan preparats per al tractament de les dades i el primer que s'ha hagut de fer és netejar els fitxers i presentar-los de manera adequada per poder tractar les dades i analitzar-les.

El treball realitzat estudia la relació entre les notes d'accés i les notes durant la fase inicial del Grau en Enginyeria en Tecnologies Industrials. És per això que des d'un principi s'ha deixat de banda el fitxer de notes de la fase no inicial, és a dir, de cursos posteriors a 1r.

A més, pel fet de tractar només amb alumnes de GETI, totes les dades personals i qualificacions d'alumnes pertanyents a GEQ o GEM també han estat descartades.

Per altra banda dins del fitxer de dades personals de l'estudiant hi ha moltes d'aquestes dades que són irrelevantes per al nostre estudi i que, per tant, també s'han descartat. És el cas del codi postal, el tipus d'accés i el centre del que provenen.

Finalment, dins del document que conté les notes de la fase inicial, tot i que és el més important per a l'estudi, també hi ha dades que no ens aporten cap informació per al nostre treball. En aquest cas les dades descartades han estat les del grup en què està matriculat l'estudiant.

Cal destacar també que no n'hi ha prou només amb aquest filtre, ja que després de tot aquest procés encara hi ha alumnes que no entren dins el nostre estudi. És, per exemple, el cas d'alumnes que han fet la preinscripció al GETI, i per tant es tenen les dades personals dels alumnes, i després no han cursat les assignatures. També s'han trobat casos en que passa tot el contrari, és a dir, que no es tenen les dades personals de l'alumne i, en canvi, sí que se'n tenen de les assignatures; això pot haver passat amb alumnes que entren per una altre via. Aquests alumnes que apareixen en només un dels dos fitxers s'han descartat de cara a l'anàlisi de les dades.

Dins del fitxer de dades que es va proporcionar des de Prisma també hi ha algun cas d'alumnes preinscrits abans del 2010, és a dir, abans de començar el Grau. Òbviament, no es pot fer un estudi entre alumnes de Grau i alumnes de Pla antic, ja que la matèria no és equiparable i, fins i tot, hi ha assignatures que canvien.

Ja per últim, s'ha detectat alumnes amb una nota d'accés inferior a 9, la qual cosa vol dir que no han entrat per la via de selectivitat habitual ja que la nota d'accés en tots aquests anys ha estat superior.

Tots aquests casos especials no s'han tingut en compte per a l'estudi i s'han hagut de filtrar minuciosament.

Un cop s'han descartat tots aquests estudiants no aptes per a l'estudi que s'ha dut a terme, s'ha procedit a decidir un criteri per aquells que sí que ho són. Amb això es vol que l'estudi sigui el més realista possible, posant les mateixes condicions per a cada un dels estudiants. És per aquesta raó que s'ha fet l'estudi usant només la nota de la primera convocatòria a cada assignatura.

Aquesta tria ha estat senzilla d'implementar ja que només es treballa amb el primer any del Grau. Per fer-ho, s'ha comprovat que l'any de matriculació coincideixi amb el de presentació a la selectivitat. A més per les assignatures del quadrimestre de tardor, s'ha hagut de tenir en compte que s'haguessin cursat durant el primer quadrimestre ja que hi ha la possibilitat de que alguns alumnes no aprovin la primera convocatòria i es presentin altra vegada al quadrimestre de primavera.

### 4.3. Presentació de les dades

Ja decidides les dades rellevants per al projecte, s'han presentat de manera adequada per poder-les tractar posteriorment. A tal fi, s'han combinat els dos documents d'interès (el de dades personals i el de notes de fase inicial) en un únic fitxer en format csv (comma-separated values). Aquest tipus de format és molt adequat en el cas que es vulgui tractar amb les dades posteriorment, ja que fa molt fàcil l'accés a les dades desitjades. A més, com s'ha vist prèviament és bàsic per poder usar el mòdul Pandas de Python.

	CODEX	SEXE	ANY	SELE	ALG	CAL	MEC	QUI	INF	GEO	CAL2	TER	\
0	277551	H	2013	12.042	6.2	7.5	6.7	6.8	7.1	7.6	5.3	5.0	
1	244848	H	2011	11.390	7.1	7.3	5.5	5.6	7.3	6.2	7.8	5.8	
2	245234	H	2011	10.240	5.3	4.5	4.8	5.3	6.1	6.4	6.2	3.4	
3	227703	D	2010	10.586	4.3	3.4	4.2	6.5	5.7	-1.0	-1.0	1.6	
4	245776	H	2011	9.774	5.0	5.9	5.7	4.2	6.6	5.2	5.5	1.9	
5	228122	H	2010	9.772	5.0	6.1	4.0	5.9	7.7	5.8	5.0	5.8	
6	245810	H	2011	11.400	6.7	7.0	7.4	6.5	5.1	8.2	8.1	7.7	
7	244262	H	2011	10.264	4.5	5.6	3.5	0.0	0.2	5.1	5.6	-1.0	
8	244287	H	2011	11.204	6.3	6.1	5.0	4.2	6.0	6.1	6.8	6.8	
9	244289	H	2011	12.008	7.0	9.0	7.4	5.3	8.5	6.5	8.3	7.0	
10	244235	H	2011	12.330	6.7	8.7	6.8	5.1	8.4	7.4	7.3	8.7	
11	294152	H	2014	12.014	5.7	5.9	5.0	6.1	7.7	3.0	3.5	5.4	
12	227686	H	2010	10.076	3.5	3.1	3.4	3.1	0.4	-1.0	-1.0	-1.0	
13	231199	H	2010	11.330	0.0	3.5	2.4	3.9	2.9	-1.0	6.5	-1.0	
14	245591	H	2011	10.294	6.1	4.4	5.5	2.5	4.3	6.3	-1.0	5.1	
15	245620	D	2011	10.374	6.0	6.6	3.2	3.6	4.0	6.0	6.3	-1.0	

Il·lustració 1. Fitxer de dades en format DataFrame

## 5. Interfície gràfica

### 5.1. PyQt

En aquest projecte s'ha treballat amb la multiplataforma Qt per a la creació de la interfície gràfica aprofitant que gaudeix del *binding* per Python, el PyQt. És precisament per això que s'ha optat per PyQt, ja que gràcies al GETI es tenien ja uns coneixements fermes de programació amb Python.

Les següents línies de codi exemplifiquen clarament quin és el funcionament del fitxer controlador, que importa la UI gràcies a la multiplataforma Qt com es pot veure a la primera línia que es mostra.

```
from PyQt4.uic import loadUi
import sys

class Main(object):
    def __init__(self):
        self.ui = loadUi('tfg.ui')
    def show(self):
        self.ui.show()

if __name__ == '__main__':
    from PyQt4 import QtGui
    app = QtGui.QApplication(sys.argv)
    main = Main()
    main.show()
    sys.exit(app.exec_())
```

Aquest és el cos bàsic d'una GUI creada amb el PyQt. Primer de tot s'importa el mòdul *loadUi*, que permet de carregar la interfície gràfica d'usuari que s'hagi creat prèviament; en aquest cas la UI anomenada '*tfg.ui*', que més endavant s'explicarà com es crea. El mòdul *sys* que s'importa tot seguit ajuda a la interacció de la UI amb l'usuari, com per exemple, com es veu en l'última línia, permet la funció de tancada de l'aplicació.

Per altra banda tenim la classe *Main* que defineix la nostra UI. En aquest cas només conté el mètode *init*, que carrega la UI prèviament creada usant el mòdul *loadUi* que ja s'ha comentat i la funció *show*. Aquesta darrera funció és la que fa visible la UI per l'usuari mitjançant la pantalla. En el cas que s'hagués picat a codi tota l'arquitectura de la UI també s'hauria d'haver inclòs dins de la classe que la defineix, però com s'explicarà tot seguit, el

Qt conté una eina que facilita molt aquesta tasca.

Finalment, les últimes línies simplement executen tot el que s'ha creat anteriorment.

## 5.2. Qt Designer

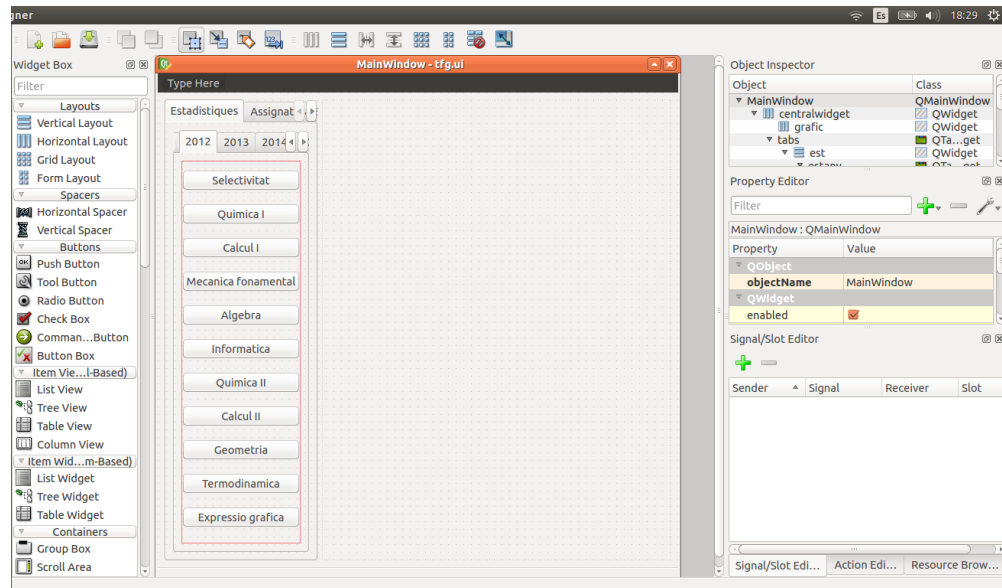
El QtDesigner és una eina pròpia de la multiplataforma Qt que permet fer un disseny d'una interfície gràfica d'usuari d'una manera molt més senzilla i intuïtiva. En comptes de picar a codi tota l'arquitectura de la interfície gràfica, permet escollir el *widget* que es vulgui i col·locar-lo on es desitgi simplement arrossegant-lo. A més d'oferir un ampli ventall de *widgets*, permet agrupar-los de diverses formes diferents i crear *layouts* amb una gran facilitat.

Per altra banda també té la opció de crear els senyals. Tot i això, en aquest projecte s'ha optat per no fer-ho així, ja que és prou senzill fer-ho des del fitxer controlador i ja es tenien els coneixements previs per fer-ho.

Per carregar el disseny fet amb el QtDesigner mitjançant els fitxers i visualitzar-lo, ja s'ha vist que no és gens complicat i simplement amb les línies de codi mostrades a l'apartat anterior n'hi ha prou.

Per altra banda, per relacionar el disseny fet amb el QtDesigner amb els fitxers i poder identificar i tractar cada un dels elements de la UI, simplement s'ha hagut de donar nom a cada un dels *widgets* i *layouts* utilitzats per fer l'arquitectura, així com a la mateixa interfície gràfica. Després, quan es vol treballar amb un dels *widgets*, *layouts* o amb la GUI, simplement s'ha de cridar pel nom de l'element en qüestió des del fitxer.





*Il·lustració 1. Disseny de la pantalla principal vista des del QtDesigner*

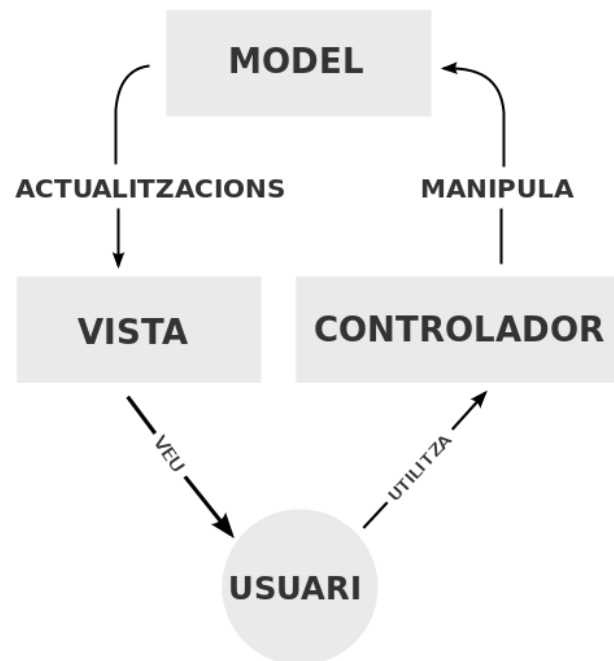
### 5.3. Model vista controlador

El model vista controlador és una estructura que s'utilitza molt per implementar el que es coneix com User Interface i és el que s'ha usat per la Graphical User Interface creada en aquest projecte. Aquest mètode ha estat bàsic per implementar l'estructura que s'ha comentat a l'apartat 3.

Gràcies al model vista controlador, ha estat molt senzill de separar les diferents parts del projecte en fitxers diferents per tal de poder-hi treballar de manera més còmoda. Aquesta estructura basa el seu funcionament en tres estacions: el model, la vista i el controlador.

- El **model** és el fitxer encarregat de fer tots els càlculs, on estan definides totes les funcions i és on recau el pes gran de l'estructura. En el nostre cas, el model és el que conté totes les funcions per tal de fer les anàlisis que es demanin en cada cas. Un cop s'han fet els càlculs i s'han executat les funcions desitjades proporciona el resultat.
- La **vista** és la part de l'estructura visible per l'usuari. A través de la vista l'usuari pot interactuar amb la interfície gràfica. En el cas d'aquest projecte, la vista permet prémer botons o lliscar per seleccionar una opció o una altra depenent de l'estudi que es vulgui.
- El **controlador**, finalment, és el que relaciona els dos blocs anteriors. El que fa

aquest fitxer és rebre els senyals de l'usuari quan interactua amb la vista i enviar un senyal al model, que s'encarrega de fer els càlculs i processos requerits per l'usuari. Un cop el model ha realitzat tots els càlculs envia el resultat al controlador i aquest és el que finalment el proporciona a la vista, on és visualitzat per l'usuari.



*II·lustració 2. Esquema del model vista controlador*

Cal destacar que tot i que teòricament el model vista controlador funciona així, també es poden aplicar canvis en funció de les necessitats i les característiques del projecte. En el nostre cas, el model no envia el resultat al controlador perquè faci els canvis pertinents a la vista, sinó que és el mateix model que canvia la vista ja sigui enviant uns resultats o afegint una gràfica.

El mètode s'ha aplicat al projecte de la següent manera:

1. El controlador recull l'acció de l'usuari i envia un senyal. En les següents línies de codi es mostren 2 alguns exemples (dos per a cada estudi) de les accions que recull el controlador. En tots aquests casos, el senyal s'envia quan es deixa anar el botó indicat en cada cas. La primera part de cada línia es refereix la interfície (*self.ui*), després indica quin és el botó que envia aquell senyal i finalment decidim quan s'envia aquest senyal; en aquest cas quan es deixa anar (*released*).

```
#estimacions
```

```
self.ui.esti0.released.connect(self.estimacions0)
```

```
self.ui.esti95.released.connect(self.estimacions95)
```

```
#grafiques
```

```
self.ui.galg15.released.connect(self.showalgebra15)
```

```
self.ui.gqui215.released.connect(self.showquimica215)
```

```
#estadistiques
```

```
self.ui.ealg15.released.connect(self.statsalgebra15)
```

```
self.ui.eSele15.released.connect(self.statssele15)
```

```
#grafiques aprovades
```

```
self.ui.t2010.released.connect(self.stack10)
```

```
self.ui.t2011.released.connect(self.stack11)
```

2. Aquests senyals executen una funció, ubicada al mateix fitxer controlador. Aquesta funció és la indicada entre parèntesi a cada un dels senyals que s'envia. Es pot veure que les funcions a continuació tenen el mateix nom que les que s'han cridat al punt 1 mitjançant el senyal. Això fa que els senyals cridin cada un, una de les següents funcions. Aquestes funcions, quan són executades criden una funció, en aquest cas ubicada al fitxer model, amb els paràmetres corresponents per a què faci els càlculs.

```
def estimacions0(self):
```

```
    model.estima(self,0,9.5)
```

```
def estimacions95(self):
```

```
    model.estima(self,9.5,10)
```

```
def showquimica215(self):
```

```
    model.addmpl(self,'QUI2',2015)
```

```
def showalgebra14(self):
```

```
    model.addmpl(self,'ALG',2014)
```

```
def statssele15(self):
```

```
    model.estadistiques(self,'SELE',2015)
```

```
def statsalgebra15(self):
```

```
    model.estadistiques(self,'ALG',2015)
```

```
def stack10(self):  
    model.aprovats(self, 2010)  
def stack11(self):  
    model.aprovats(self, 2011)
```

3. Finalment el fitxer model executa la funció que se li ha demanat amb els paràmetres adequats. El resultat, segons sigui la funció que s'executa, es mostra directament a la vista mitjançant la creació d'una gràfica o un *QTextEdit*, un *widget* que permet escriure-hi dins.

## 6. Estudis

A l'apartat 5 s'han vist exemples de com funciona el model vista controlador amb els diferents estudis que s'han fet. Els estudis han estat els següents.

### 6.1. Estadístiques de les assignatures

El primer estudi del treball han estat les estadístiques de cada una de les assignatures de la fase inicial des que va començar el GETI. A més, també s'ha volgut visualitzar aquest estudi de la millor manera possible mitjançant la interfície gràfica creada amb el PyQt.

Aquest estudi s'ha fet a partir del fitxer de dades en format csv, un cop filtrades totes les dades necessàries. Un cop aquí s'ha convertit el fitxer, gràcies al mòdul Pandas, al format DataFrame per poder accedir a les dades amb més facilitat.

Amb les dades en el format DataFrame és més senzill tractar-les i, filtrant el fitxer per assignatures s'ha fet un estudi de varies dades estadístiques per cada any de matriculació. És a dir, per cada assignatura, s'ha mirat cada alumne que l'ha cursada, i en l'any que ho ha fet. Si ho ha fet en l'any que es consulta, es té en compte aquella dada i si l'ha cursada un altre any, no es comptabilitza.

Quan s'han tingut les notes de tots els alumnes que han cursat l'assignatura durant l'any de la consulta, s'ha fet un estudi estadístic en el que es calcula: la mitjana, els quartils, la desviació, màxim i mínim.

Fet l'estudi, només falta tractar amb la interfície gràfica PyQt per poder visualitzar aquest estudi. Per fer-ho, dins la finestra dels resultats s'ha creat un *widget* del tipus *QTextEdit*, que permet escriure a dins un text qualsevol.

La part del codi corresponent a aquest estudi és la següent:

```
def estadistiques(self,assignatura,curs):  
    df = pd.read_csv('final2.csv')  
    df = df[df['ANY']==curs]  
    #estadistiques sele
```

```

if assignatura=='SELE':
    x = df[df['SELE']>9]
    x = x.describe()
    x = x['SELE']
    l=['alumnes: ' +str(x[0]), 'mitjana: ' +str(x[1]), 'desviacio: ' +str(x[2]), 'min:
'+str(x[3]), '25%: ' +str(x[4]), '50%: ' +str(x[5]), '75%: ' +str(x[6]), 'max: ' +str(x[7])]
    s=assignatura
    for e in l:
        s = s + '\n' + e
#estadistiques assignatures
else:
    x = df[df[assignatura]!=-1]
    x = x[x['SELE']>9]
    x = x[assignatura].describe()
    l=['alumnes: ' +str(x[0]), 'mitjana: ' +str(x[1]), 'desviacio: ' +str(x[2]), 'min:
'+str(x[3]), '25%: ' +str(x[4]), '50%: ' +str(x[5]), '75%: ' +str(x[6]), 'max: ' +str(x[7])]
    s=assignatura
    for e in l:
        s = s + '\n' + e

clean_layout(self.ui.graficlayout)
fig = QTextEdit()
fig.setText(s)
self.ui.graficlayout.addWidget(fig)

```

Si s'analitza una mica el codi es pot veure que, primerament es llegeix el fitxer de dades i es converteix a format DataFrame. Posteriorment es selecciona quines són les dades d'interès, depenent dels inputs que s'hagin rebut i es procedeix a l'anàlisi.

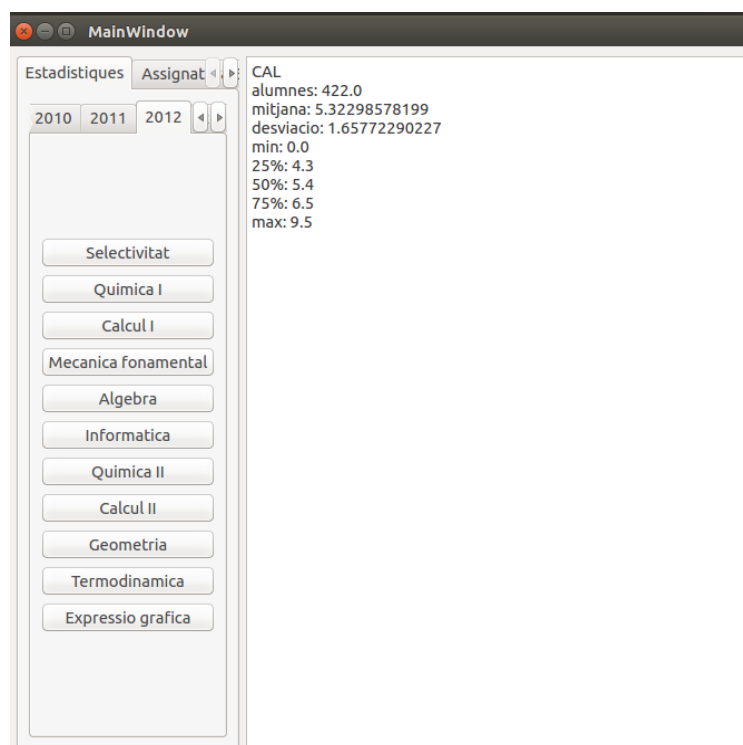
Per fer l'anàlisi s'ha separat la funció entre l'estudi per selectivitat o per la resta d'assignatures. Aquest fet és perquè la condició posterior per la selecció de dades és diferent en els dos casos. Quan tractem dades de la selectivitat, tan sols volem aquelles que són per sobre de 9, mentre que quan estudiem les altres assignatures, no volem tenir en compte aquelles que tenen un -1 de valor, és a dir, que l'alumne en qüestió no ha cursat

aquella assignatura.

Un cop separats aquests dos blocs amb un *'if'* i seleccionades les dades d'interès es fan els càlculs pertinents. Els *DataFrame* tenen una funció molt útil per quan es tracta d'estudis estadístics que es el mètode *describe*. Aquest mètode retorna directament diverses mesures estadístiques (màxim, mínim, mitjana, desviació i quartils).

Per mostrar les dades obtingudes amb la UI, s'ha creat un *widget* del tipus *QTextEdit*. Aquest *widget* només accepta *strings* i és per això que s'han convertit totes les dades obtingudes a aquest format. A més, s'ha escrit el nom de cada mesura estadística davant per poder-les visualitzar amb claredat.

Tot i això, abans de mostrar les dades a la UI, s'ha hagut d'implementar el mètode *clean\_layout*, que neteja la finestra que se li digui per poder introduir nous *widgets*.



Il·lustració 3. Un dels molts resultats de l'estudi d'estadístiques, en aquest cas de càlcul de l'any 2012.

## 6.2. Gràfica notes vs selectivitat

En aquest cas, l'estudi ha estat la relació de les notes de cada assignatura amb les notes de selectivitat de cada alumne que ha cursat aquella assignatura l'any de la consulta. Per visualitzar l'estudi, s'ha fet mitjançant una gràfica i la seva recta de regressió dins la interfície PyQt.

Per realitzar aquesta part del treball també s'ha usat des d'un bon principi el mòdul Pandas a fi de convertir el fitxer que conté les dades en un DataFrame. Un cop s'han tingut totes les dades al DataFrame, s'ha filtrat el fitxer de la mateixa manera que en l'estudi anterior, és a dir, per assignatures. S'han agafat les dades de tots els alumnes que han cursat l'assignatura consultada l'any desitjat per poder extreure la nota que han tret en aquella assignatura i la nota d'accés que tenen.

Amb aquestes dues dades ja es pot realitzar la gràfica que relaciona les notes de l'assignatura desitjada amb les notes d'accés dels alumnes. Aquesta gràfica s'ha fet mitjançant el mòdul Matplotlib, que com ja hem comentat, és el mòdul que s'ha usat per realitzar les gràfiques en aquest treball.

Per altra banda, s'ha calculat la recta de regressió de la gràfica amb la funció *polyfit* del mòdul Numpy per poder veure més clarament quina és la relació de les notes de l'assignatura amb la nota d'accés.

Un cop fet aquest estudi només resta fer la visualització. Per fer-ho la finestra de resultats s'ha dividit en dos seccions diferenciades: una per a la gràfica i l'altre per mostrar la recta de regressió del mateix. Per poder visualitzar la gràfica ha estat necessari crear un *widget* del tipus *FigureCanvas*, mentre que per mostrar la recta de regressió s'ha usat una altra vegada el *widget* del tipus *QTextEdit*.

El tipus de *widget* *FigureCanvas* permet d'introduir-hi una figura qualsevol, que en el nostre cas ha estat la gràfica creat gràcies al Matplotlib. Per altra banda, el *QTextEdit* ens permet d'escriure un text per mostrar-lo a la finestra on l'ubiquem.

Per veure més clarament quin ha estat el camí per fer aquest estudi, el codi és el següent



```
def rectaregressio(assig,curs):
    df = pd.read_csv('final2.csv')
    df = df[df['SELE']>9]
    df = df[df['ANY'] == curs]

    data = df[df[assig]!=-1]
    data = data['SELE']
    data2 = df[df[assig]!=-1]
    data2 = data2[assig]
    recta = polyfit(data,data2,1)

    n = ""
    for e in str(recta[0]):
        if len(n)<5:
            n = n + str(e)
    m = ""
    for i in str(recta[1]):
        if len(m)<5:
            m = m + str(i)
    recta = [m,n]
    return recta

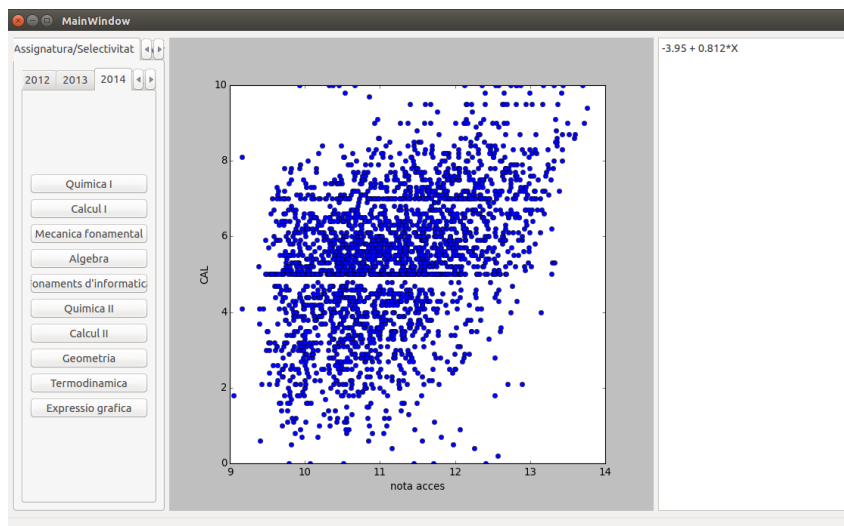
def addmpl(self, assig,curs):
    clean_layout(self.ui.graficlayout)
    df = pd.read_csv('final2.csv')
    x = df[df[assig]!=-1]
    x = x[x['SELE']>9]
    fig = Figure()
    ax1f1 = fig.add_subplot(111)
    ax1f1.plot(x['SELE'], x[assig],'o')
    self.ui.canvas = FigureCanvas(fig)
    self.ui.graficlayout.addWidget(self.ui.canvas)
    self.ui.canvas.draw()
    regressio = rectaregressio(assig,curs)
```

```

text = str(regressio[0])+' '+'+'+' + str(regressio[1])+'*'+ 'X'
recta = QTextEdit()
recta.setText(text)
self.ui.graficlayout.addWidget(recta)

```

Com es pot veure en aquest codi s'ha separat en dues funcions diferents: una per fer la recta de regressió i l'altra per crear la gràfica i afegir tant la gràfica com la recta a la UI. La metodologia és l'explicada anteriorment, però cal també destacar l'aparició altra vegada del mètode *clean\_layout*, que com ja s'ha comentat, permet de netejar la finestra abans de col·locar-hi un altre *widget* i que no hagin de coexistir els dos.



Il·lustració 4. Gràfica de la nota de càlcul de l'any 2014 contra la nota d'accés

### 6.3. Assignatures aprovades per any acadèmic

En tercer lloc, s'ha fet un estudi de quantes assignatures aproven els alumnes durant un any a l'escola. Per poder comparar els estudiants en les mateixes condicions s'ha hagut de decidir un criteri. Tot i que és possible fer el mateix estudi amb altres criteris, s'ha decidit considerar el primer any acadèmic de cada alumne a l'ETSEIB. Amb aquesta condició, el màxim nombre d'assignatures que es poden aprovar són 10, per tant, de cada estudiant, es mira quantes n'ha aprovat i se'l posa dins del bloc que li pertoca en funció de quantes són les assignatures aprovades.

Un cop s'ha realitzat l'estudi i tots els alumnes són dins del bloc corresponent s'ha hagut de pensar la manera de visualitzar-ho per l'usuari. Com en els casos anteriors l'estudi s'ha dividit per any acadèmic; és a dir, els que van començar el Grau un any o un altre. I dins de cada any el que s'ha creat és una gràfica de barres que ens il·lustra de manera molt clara el nombre d'assignatures que aproven els estudiants.

Part del codi que s'ha usat per fer l'estudi és el següent:

```
def aprovats(self,curs):
    f = open('final2.csv','r')
    aprova0 = 0
    aprova1 = 0
    aprova2 = 0
    for e in f:
        e = e.strip()
        s = e.split(',')
        if s[0]!='CODEX' and float(s[2]) == curs:
            n = 0
            for i in range(4,14):
                if float(s[i])>5:
                    n = n + 1
            if n == 0:
                aprova0 +=1
            if n == 1:
                aprova1 +=1
            if n == 2:
                aprova2 +=1

A = [aprova0]
B = [aprova1]
C = [aprova2]

clean_layout(self.ui.graficlayout)
```

```
fig = Figure()

g = fig.add_subplot(111)

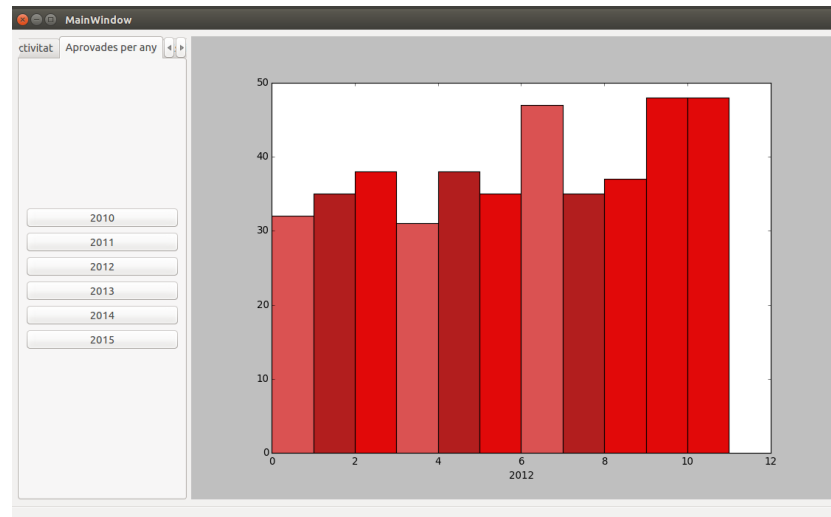
g.bar(0, A, color = '#DA5252',width = 1)
g.bar(1, B, color = '#B21E1E',width = 1)
g.bar(2, C, color = '#E10909',width = 1)

self.ui.canvas = FigureCanvas(fig)
self.ui.graficlayout.addWidget(self.ui.canvas)
self.ui.canvas.draw()
```

No cal fixar-se gaire en el codi per donar-se compte que només hi ha una part. En aquest fragment només es mostren els que aproven 0, 1 o 2 assignatures, però la resta del codi seria anàloga a aquesta part.

Primer de tot, s'han creat unes variables de nombre enter per anar comptabilitzant el nombre d'alumnes en cada bloc d'assignatures aprovades. Fet això, s'ha llegit el fitxer per files per poder tractar amb la informació en funció de l'alumne. Per accedir al fitxer per files, ha estat més fàcil de no tractar-lo com a DataFrame i fer-ho com un fitxer de text. Posteriorment, s'ha hagut de modificar una mica el format de cada fila però s'ha pogut tractar bé cada alumne i col·locar-lo al bloc corresponent.

Amb les dades ben tractades, fer la gràfica ja és possible i, gràcies al Matplotlib s'ha creat una gràfica de barres molt visual. Finalment, amb un *widget* del tipus *FigureCanvas*, s'ha afegit la gràfica a la UI. Cal destacar també el mètode *clean\_layout*, per netejar la finestra on s'ha de col·locar la gràfica abans d'afegir-lo.



Il·lustració 5. Gràfica que mostra el nombre d'assignatures aprovades pels alumnes l'any 2012

## 6.4. Predicció

Per últim s'ha realitzat una estimació de les notes que podria treure un alumne que entra a l'escola per fer el Grau en Enginyeria en Tecnologies Industrials basat en les estadístiques d'alumnes anteriors sabent la seva nota d'accés.

L'estudi determina la nota que podria treure un alumne qualsevol a cada una de les assignatures de fase inicial. Per fer-ho, l'estudi es basa en totes les persones que han tret una nota d'accés similar a la de l'alumne en qüestió, fent una mitjana de totes elles.

Per poder fer aquest estudi s'ha procedit inicialment com en els altres: creant el DataFrame contenint totes les dades. Un cop les dades han estat agrupades al DataFrame s'han filtrat segons la nota d'accés. Per filtrar-les s'ha hagut de decidir un criteri, i s'han fet grups d'alumnes que tenen la nota compresa en un rang de mig punt. Seguint aquest criteri, s'ha fet els següents grups: una nota d'accés inferior a 9,5, entre 9-9,5, de 9,5 a 10, entre 10 i 10,5, 10,5-11, 11-11,5, 11,5-12, 12-12,5, 12,5-13 i superior a 13.

Per dur a terme aquest estudi s'ha usat el codi següent:

```
def estima(self,nmin,nmax):
    df = pd.read_csv('final2.csv')
```

```
df = df[df['SELE']>nmin]
df = df[df['SELE']<nmax]

calcul = df[df['CAL']!=-1]
calcul = calcul['CAL'].mean()

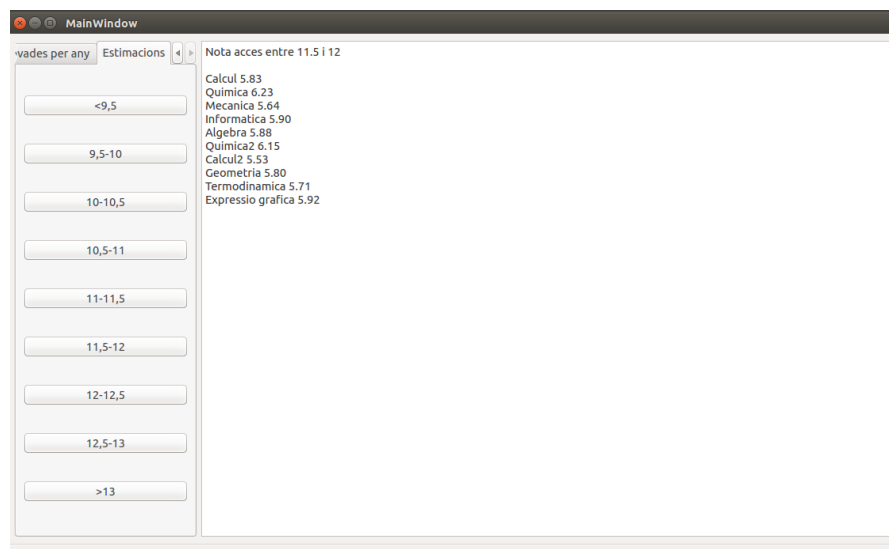
scalcul = ""
for e in str(calcul):
    if len(scalcul)<4:
        scalcul = scalcul + str(e)
s = 'NOTES' + '\n' + '\n'
s = s + 'Calcul ' + scalcul + '\n'

clean_layout(self.ui.graficlayout)
notes = QTextEdit()
notes.setText(s)
self.ui.graficlayout.addWidget(notes)
```

Aquest codi, com en el cas de les gràfiques acumulats, només és una part, ja que la resta del codi és anàloga a aquesta. Com en tots els casos, s'ha començat filtrant el fitxer de dades per les que són d'interès. En aquest cas, les dades que ens interessen són les que estan dins el rang de notes d'accés que l'usuari desitja.

A partir de tots aquests alumnes que estan dins del rang de notes d'accés requerida, s'ha fet la mitjana de cada una de les assignatures aprofitant la funció *mean* dels DataFrame. Un cop tenim la nota que s'espera que traurà un alumne dins aquell rang, fem la visualització.

Per a què sigui més còmode la visualització s'ha limitat del nombre de decimals a 2, mitjançant un *for*. A continuació s'ha transformat la mitjana a format *string* per poder inserir-la dins del *widget* del tipus *QTextEdit* i afegir-la directament a la UI. Abans d'afegir-lo, però, s'ha hagut de netejar la pantalla amb el mètode *clean\_layout*.



Il·lustració 6. Estimació de les notes que trauran els alumnes amb una nota d'accés entre 11,5 i 12





## 7. Impacte ambiental

L'impacte ambiental d'aquest projecte es deu, principalment, a les emissions de CO<sub>2</sub> provinents de:

- La generació d'energia elèctrica necessària per la il·luminació i pel funcionament dels equips utilitzats en la seva realització.
- La fabricació del paper consumit.

Si desglossem el consum de l'energia elèctrica durant la realització del projecte tenim:

Element	Potència (W)	Temps (h)	Consum (kWh)
Ordinador propi	75	350h	26,25
Bombeta halògena	70	350h	24,5
		<b>Total</b>	<b>50,75</b>

Taula 1. Consum d'energia elèctrica

La Oficina Catalana del Canvi Climàtic de la Generalitat de Catalunya (2016) estima el mix de producció bruta de la xarxa elèctrica peninsular en 300 g CO<sub>2</sub>/kWh.

Prenent aquest valor, les emissions són: 50,75 kWh x 0,3 kg CO<sub>2</sub>/kWh = 15,23 kg CO<sub>2</sub>

Per altra banda, tenim que el consum de paper ha estat aproximadament de 100 fulls DIN A4 no reciclat.

Segons la Confederació de Comerç de Catalunya (2010), el factor d'emissió per fabricar 2500 fulls DIN A4 de paper no reciclat es pot estimar en 36 kg CO<sub>2</sub>. Això vol dir 1,44 kg CO<sub>2</sub> per cada 100 fulls.

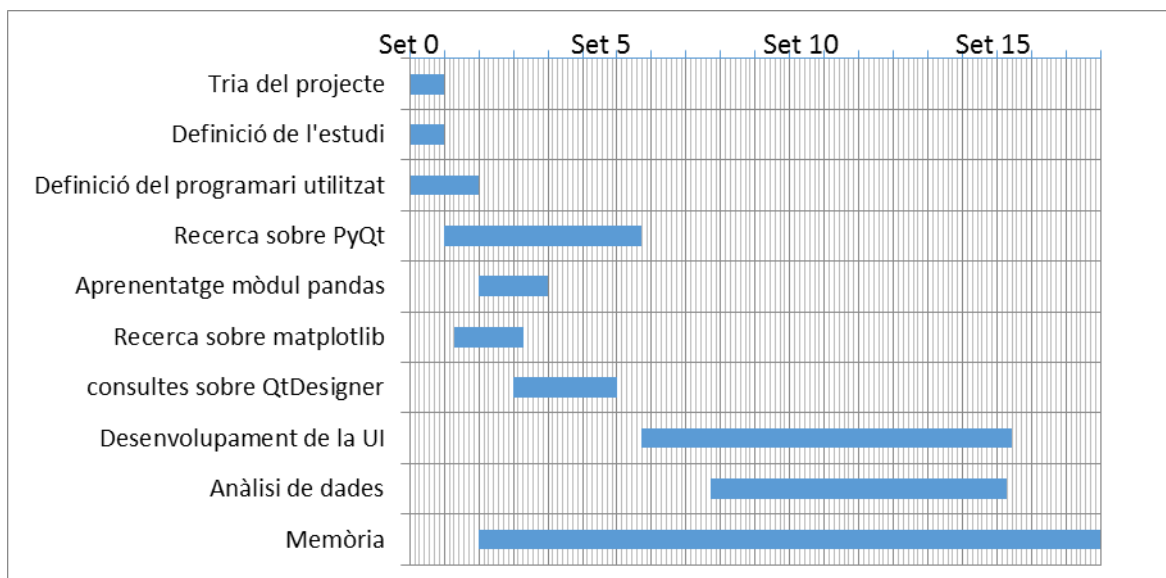
### Emissions totals

Si considerem ambdós factors, les emissions totals de CO<sub>2</sub> a causa del projecte han estat de 1,44 + 15,23 = **16,67 kg CO<sub>2</sub>**

## 8. Planificació i valoració econòmica

### 8.1. Planificació

La durada total del projecte ha estat de 4 mesos i mig, començant la primera setmana de febrer fins la tercera de juny inclosa. Això fa un total de 20 setmanes que s'han distribuït de la següent manera:



*II·lustració 7. Diagrama de Gantt que mostra clarament la planificació del projecte*

De les tasques indicades al diagrama de Gantt de la il·lustració 7 en podem fer 4 grans grups. Aquests grups són, per una banda, la tria del projecte i la definició del mateix, per decidir exactament l'objectiu i l'abast del projecte.

Tot seguit, s'ha procedit a la recerca i aprenentatge sobre diferents temes. S'ha hagut de dedicar moltes hores a la recerca sobre el funcionament de la multiplataforma Qt, ja que abans de començar el projecte en desconeixia totalment el funcionament. A més, s'ha hagut de consultar repetidament, sobretot al principi com indica el diagrama, les documentacions dels mòduls pandas i matplotlib. Tot i que la gran part de recerca i aprenentatge va ser durant la fase inicial, ha estat present puntualment durant tot el projecte, ja que sempre que s'ha tingut un dubte s'ha consultat a les pàgines web pertinents.

Un cop assolits els coneixaments bàsics, s'ha procedit a començar el tercer gran bloc del treball: el desenvolupament de la UI i de les funcions per a l'anàlisi de dades. Es va decidir

de començar la creació de la UI abans de treballar amb l'anàlisi de dades perquè així, a mesura que s'ha anat avançant amb l'anàlisi de dades s'ha pogut visualitzar clarament mitjançant la UI prèviament creada. Tot i això, s'han acabat de treballar les dues coses en paral·lel fins els últims detalls.

L'últim gran bloc del projecte és el de la memòria. Aquest és un apartat que tot i que s'anomena el darrer ha estat present durant tota la durada del projecte. A mesura que s'ha anat avançant la part pràctica del treball amb el desenvolupament de la UI i l'anàlisi de dades s'ha anat especificant a la memòria fins l'últim dia.

## 8.2. Valoració econòmica

La valoració econòmica d'aquest treball només té dos factors en compte. Per una banda, l'esforç realitzat per dur a terme el projecte, expressat en hores de treball, i per altra banda els equips informàtics necessaris pel treball, que inclouen un ordinador i una impressora.

Element	Quantitat	Cost unitari	Cost total
Enginyer industrial	350h	40€/h	14000€
Ordinador	1	600€	150€
Impressora	1	70€	70€
		<b>Total</b>	<b>14220€</b>

*Taula 2. Cost del projecte*

## Conclusions

Una base de dades té poca utilitat si no es sap tractar amb aquestes dades i fer-ne un bon estudi que sigui clar i entenedor per poder veure quin és el comportament d'aquell sistema. És per això que en aquest treball s'ha buscat de fer tot aquest post-procés un cop es té una base de dades emmagatzemada.

En el cas d'aquest treball s'ha agafat com a base de dades les notes dels estudiants del Grau en Enginyeria en Tecnologies Industrials, però la base de l'estudi seria la mateixa per a qualssevol altre estudi que es volgués fer en un futur.

Per una banda, si amb aquesta base de dades es volgués fer un tipus d'estudi diferent no hi hauria cap complicació, ja que la base de dades ja està filtrada. A més la UI també està creada i només faltaria afegir-hi les funcions que fessin possible aquest nou estudi.

Per altra banda, si es volgués fer un estudi a partir d'una base de dades diferent, simplement s'hauria de canviar el disseny de la UI si es creu necessari. Evidentment, també s'hauria de canviar la base de dades, però l'esquema no canviaria gaire i, seguint el guió d'aquest treball, seria ben senzill.

També cal destacar la flexibilitat en la vista gràcies al model vista controlador, que separa molt bé el que és la vista i el model. És a dir, com que tenim molt identificat el que pertany a la UI i el que pertany a l'anàlisi de dades, es podria canviar la vista per, per exemple, poder-ho visualitzar des d'un dispositiu mòbil.

Pel que fa al resultat dels estudis realitzats, es constata en tot moment que la relació entre nota d'accés i notes a la fase inicial és positiva, és a dir, que a més nota d'accés, millors notes es trauran teòricament a la fase inicial, com ja era d'esperar.

Com a resultats interessants cal destacar que mentre que per a les persones amb una nota d'accés baixa l'assignatura de fonaments d'informàtica és una de les que costa més, per a les persones amb una nota d'accés més alta, és de les assignatures que els va millor. Química és normalment l'assignatura que va millor i càlcul 2 i mecànica les que costen més generalment.

Si es miren les assignatures aprovades en un any acadèmic, es pot veure que durant els dos primers anys de grau hi va haver molta gent que va aprovar-les totes, mentre que més tard s'ha anat equilibrant.

Comprovant els resultats de l'estudi que proporciona la recta de regressió de la gràfica

entre les notes de cada assignatura i la nota d'accés, és clar que l'assignatura que proporciona una recta de regressió amb el pendent més elevat és la de fonaments d'informàtica. D'això se'n conclou directament que, a la que augmenta la nota d'accés de l'alumnat, la nota que es veu més afectada és la de fonaments d'informàtica.

Aquests resultats que afecten a l'assignatura de fonaments d'informàtica es poden deure a que és una assignatura que, tot i no ser complicada, a les escoles no s'ensenya. Això fa que un cop a la universitat tothom comença de zero i la gent que té més capacitat o hi posa més ganes entén el llenguatge informàtic abans.

Finalment, es pot concloure que els objectius han estat assolits al crear una interfície gràfica còmode per l'usuari que permet visualitzar clarament els resultats de l'estudi proposat. A més, aquest estudi pot ser un propulsor per estudis futurs i facilitar molt el post-procés de la base de dades oferint una visualització molt clara i entenedora.

## Agraïments

Vull donar les gràcies al meu tutor Lluís Solano per l'ajuda i el consell en tot moment. Sempre ha tingut una visió crítica molt positiva del treball. A més, la seva ajuda va ser fonamental al principi a l'hora d'enfocar clarament l'objectiu del projecte.

Un agraïment d'allò més especial per la Marta, per totes les correccions, crítiques i opinions, però sobretot per fer-me tan feliç.

També, com no, agrair el suport incondicional de la família més propera, els meus pares i germana que, sense ser-ne conscients, m'ajuden tant a tirar endavant cada dia.

I, finalment, el meu més sincer agraïment als meus avis, Emili i Roser, per l'alegria i la vitalitat que contagien, pel suport que donen i l'amor que transmeten.

## Bibliografia

- ANDRE, M. *PyQt Tutorials*. [en línia] [Consulta: Març 2016] Disponible a:  
<https://wiki.python.org/moin/PyQt/Tutorials>
- CONFEDERACIÓ DE COMERÇ DE CATALUNYA. *Pla de reducció d'emissions de CO2*, Barcelona, 2010. P. 14 [Consulta: Juny 2016] <http://www.confecom.cat/wp-content/uploads/2008/04/PLA-REDUCCIO-EMISSION-CO2-CCC.pdf>
- MATPLOTLIB. *Bar graph*. [en línia] [Consulta: Maig 2016] Disponible a:  
[http://matplotlib.org/examples/pylab\\_examples/barchart\\_demo.html](http://matplotlib.org/examples/pylab_examples/barchart_demo.html)
- MATPLOTLIB. *Installation*. [en línia] [Consulta: Març 2016] Disponible a:  
<http://matplotlib.org/users/installing.html>
- MATPLOTLIB. *Tutorial*. [en línia] [Consulta: Març 2016] Disponible a:  
[http://matplotlib.org/users/pyplot\\_tutorial.html](http://matplotlib.org/users/pyplot_tutorial.html)
- NELSON, R. *PyQt Tutorial*. [en línia] [Consulta: Febrer 2016] Disponible a:  
<http://blog.rcnelson.com/building-a-matplotlib-gui-with-qt-designer-part-1/>
- NUMPY. *Polyfit*. [en línia] [Consulta: Maig 2016] Disponible a:  
<http://docs.scipy.org/doc/numpy/reference/generated/numpy.polyfit.html>
- OFICINA CATALANA DEL CANVI CLIMÀTIC. *Nota informativa sobre la metodologia d'estimació del mix elèctric per part de l'oficina catalana del canvi climàtic (occc)*, Catalunya, 2016. [Consulta: Juny 2016]  
[http://canviclimatic.gencat.cat/web/.content/home/reduex\\_emissions/factors\\_emissio\\_associats\\_energia/160219\\_Nota-metodologica-mix\\_cat.pdf](http://canviclimatic.gencat.cat/web/.content/home/reduex_emissions/factors_emissio_associats_energia/160219_Nota-metodologica-mix_cat.pdf)
- PANDAS. *DataFrame documentation*. [en línia] [Consulta: Abril 2016] Disponible a:  
<http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html>
- PANDAS. *Installation*. [en línia] [Consulta: Març 2016] Disponible a:  
<http://pandas.pydata.org/pandas-docs/stable/install.html>
- PANDAS. *Tutorial*. [en línia] [Consulta: Març 2016] Disponible a:  
<https://vimeo.com/59324550>

- STACK OVERFLOW. *Python. Mètode join*. [en línia] [Consulta: Febrer 2016] Disponible a: <http://stackoverflow.com/questions/1876191/explain-python-join>
- THE QT COMPANY LTD. *Qt Documentation. Signals*. [en línia] [Consulta: Març 2016] Disponible a: <http://doc.qt.io/qt-5/signalsandslots.html>
- THE QT COMPANY LTD. *Qt documentation. Widgets and Layouts*. [en línia] [Consulta: Març 2016] Disponible a: <http://doc.qt.io/qt-4.8/widgets-and-layouts.html>
- THE QT COMPANY LTD. *Qt Installation*. [en línia] [Consulta: Febrer 2016] Disponible a: <http://doc.qt.io/qt-5/linux.html>
- THE QT COMPANY LTD. *Qt Documentation*. [en línia] [Consulta: Febrer 2016] Disponible a: <http://doc.qt.io/qt-4.8/>
- THE QT COMPANY LTD. *QtDesigner manual*. [en línia] [Consulta: Abril 2016] Disponible a: <http://doc.qt.io/qt-4.8/designer-manual.html>
- W3SCHOOLS. *Hexadecimal colors*. [en línia] [Consulta: Maig 2016] Disponible a: [http://www.w3schools.com/colors/colors\\_picker.asp](http://www.w3schools.com/colors/colors_picker.asp)